Name: Pearl Law
SCU ID: W0958839
Rank: 7
F1-score: 0.8673

Movie Review Sentiment Classification with Convolutional Neural Networks and Long Short-Term Memory

Abstract

Important features such as emotions can be derived from textual data. Sentiment classification is a common natural language processing problem that allows text data to be labeled into positive, neutral, or negative categories. Support vector machines and recurrent neural networks (RNN) are popular machine learning algorithms used for sentiment analysis. In this project, we explore the effectiveness of long short-term memory (LSTM) networks, a type of RNN, for sentiment classification of movie reviews.

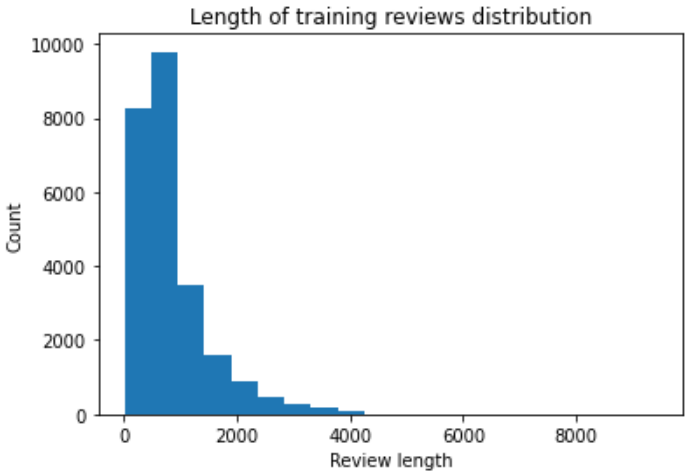
Introduction

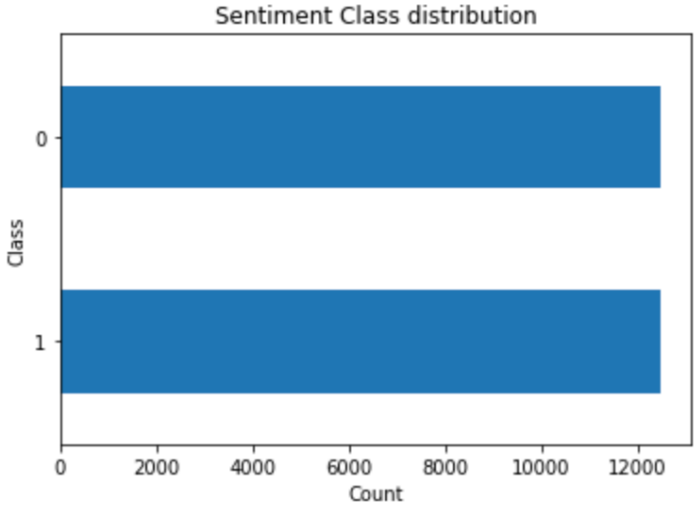
Sentiment analysis is a commonly used technique to evaluate positive, neutral, or negative feelings on a particular product or experience through text data. RNNs are useful for classification tasks where the input data is in sequential or time-series format. In the case of lengthy sequence problems, however, RNNs encounter a vanishing gradient problem that cannot be resolved other than by changing the existing RNN architecture. As a result, LSTMs were introduced to deal with the vanishing gradient problem seen in large sequence datasets. LSTMs are also desirable for preserving information over many timesteps. The goal of this project is to implement an LSTM neural network to predict whether a given movie review has positive or negative sentiment.

Approach

The first step to optimizing the classification of movie reviews based on positive or negative sentiment is to preprocess each text review in the training and test datasets and remove unnecessary words or noise that could negatively impact the model's performance. Every word in a review is converted to lowercase. All HTML tags are removed and a list of English stop words from the Natural Language Toolkit (NLTK) corpus is used to remove text elements that are not useful for classification. Contracted words are expanded to their full form to help derive meaning between words during the word embedding process in later steps. For example, "couldn't" is expanded to "could not" to obtain meaning from "not" that otherwise would be difficult. Punctuations, and accents on alphabet characters are all removed to enhance key features within each movie review. Finally, negative sentiment labeled as -1 are encoded as 0 to utilize sigmoid activation function for binary classification.

Examining the word length of reviews for the training and test datasets is necessary to understand the word length distribution for the entire datasets. The maximum word length of a review is 9425 and 8869 for train and test datasets, respectively, and the average word length of a review is 845 and 855 for train and test datasets, respectively.





Before feeding the data into the LSTM model, the training data is tokenized and vectorized with 2000 features with maximum length = 800 (based on the average length of most reviews in training and test datasets). This preprocessing layer computes a vocabulary of string terms from the tokens in the training dataset only.

The neural network model contains the following layers:

- Preprocessing layer that acts as an encoder to convert each review to a sequence of token indices
- Embedding layer that takes input sequences of length 800 and converts 2000 features to a dense vector of length 64
- Bidirectional LSTM layer with 64 neurons

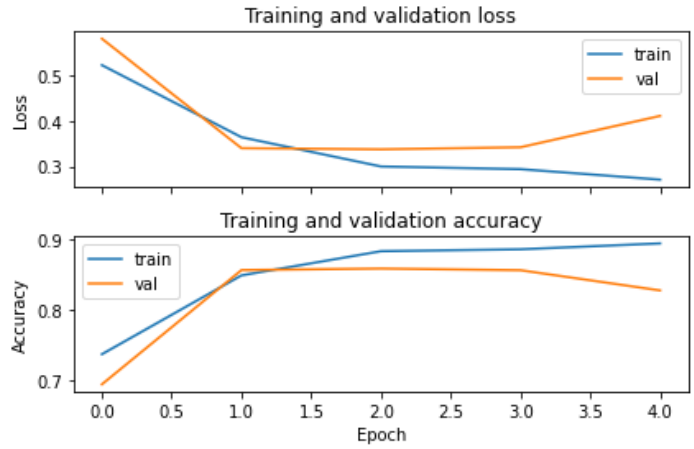
- Dense, fully connected output layer with 1 neuron and sigmoid activation function to predict whether a given review is positive (1) or negative (0)

The model is compiled with binary cross entropy loss and Adam optimizer (lr = 1e-3) and trained using batch size of 32 for 10 epochs. Early stopping is implemented to stop training when there is no further improvement in validation loss after 2 consecutive epochs. 20% of the training data was held out for validation.

Results

After training, we observe the following training and validation loss and accuracy below.

```
782/782 [==============================] - 24s 31ms/step - loss: 0.2794 - accuracy: 0.8934
Training loss: 0.2793751060962677, Training accuracy: 0.8934000134468079
```



We can see that training loss decreases and accuracy increases with each epoch. Validation loss decreases up to the 3rd epoch, then appears to increase again, while the validation accuracy increases up to the 3rd epoch. This indicates that up to the 3rd epoch, the model displays a relatively good bias-variance tradeoff, but after the 3rd epoch, the model starts to perform worse. This may be due to LSTM being unstable and prone to overfitting quickly.

The best weights from training the LSTM model are used to predict the sentiment for the movie reviews in the test dataset. The negative predictions outputted by the network are decoded back to -1. The F1-score calculated on 50% of the data is 0.8673. This demonstrates that the LSTM model implemented was able to learn important textual features within the training dataset and translate that to accurately predict the sentiment of most unseen movie reviews.

Conclusion

Between traditional RNNs, LSTMs, and GRUs, we found greatest success applying LSTM in this model. Bidirectional LSTM resulted in slower training time, but better performance than unidirectional LSTM. Adding dropout did not improve the F1-score significantly. The model

architecture used in this project yielded an F1-score of 0.8673, indicating that our LSTM model was able to successfully predict movie review sentiment.